PASSED

Computational Statistics Lab5

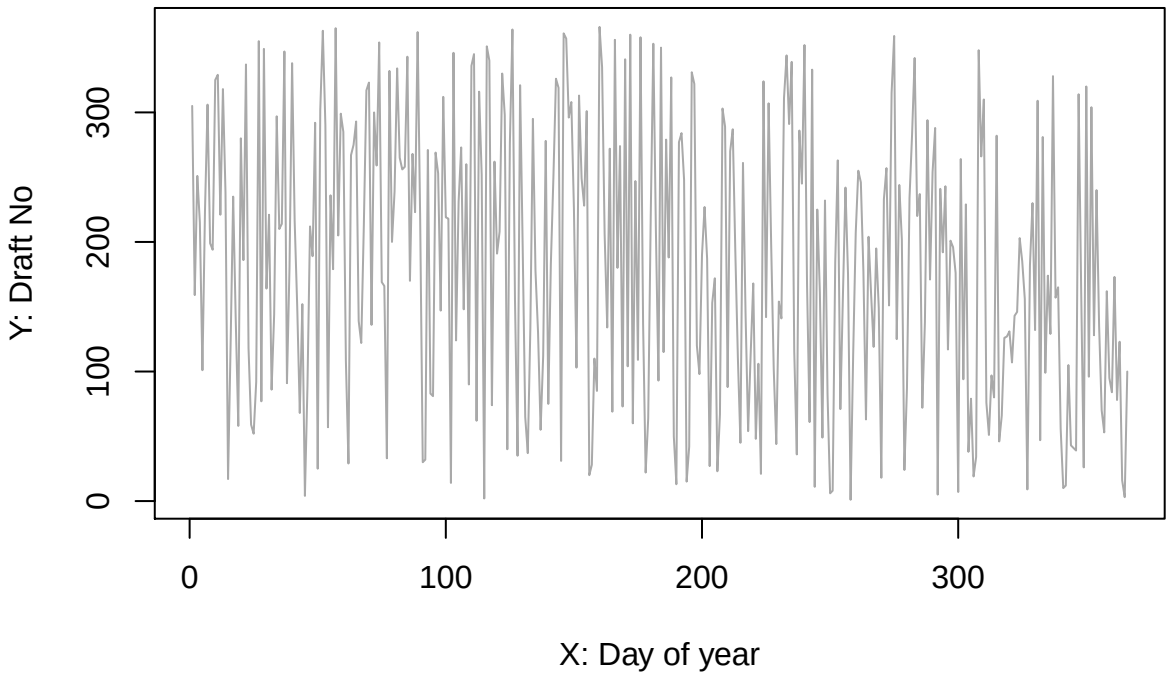
Group 6: Araya Eamrurksiri and Oscar Pettersson

March 8, 2016

Assignment1: Hypothesis testing

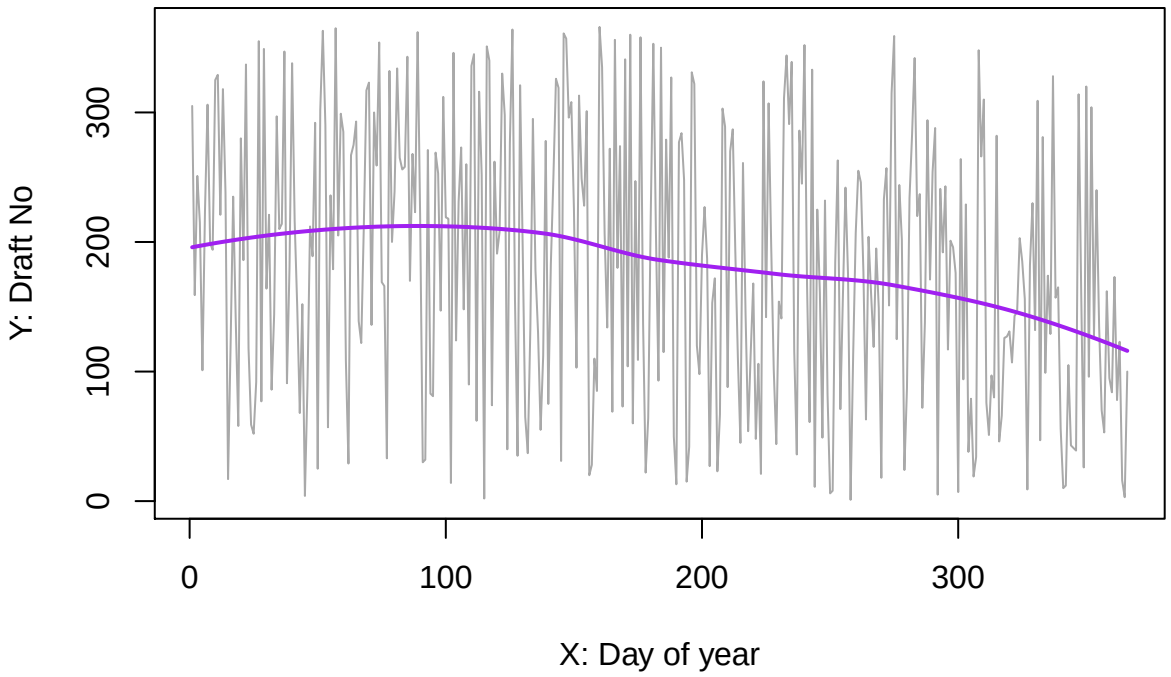
In 1970, the US Congress instituted a random selection process for the military draft. All 366 possible birth dates were placed in plastic capsules in a rotating drum and were selected one by one. The first date drawn from the drum received draft number one, the second date drawn received draft number two, etc. Then, eligible men were drafted in the order given by the draft number of their birth date. In a truly random lottery there should be no relationship between the date and the draft number. Your task is to investigate whether or not the draft numbers were randomly selected. The draft numbers (Y=Draft_No) sorted by day of year (X=Day_of_year) are given in the file lottery.xls

1.1 Make a scatterplot of Y versus X and conclud whether the lottery looks random.



After examining this plot, it can be seen that there is no obvious pattern for this data. The order of Y doesn't seem to depend on X very much. Therefore, lottery seems to be randomly selected.

1.2 Compute and estimate $\tilde{Y}$ of the expected response as a function of X by using a loess smoother (use loess()), put the curve $\tilde{Y}$ versus X in the previous graph and state again whether the lottery looks random



It does not seem that lottery is random as the computed model (indicates by purple line) is showing some kind of pattern. The value estimate $\tilde{Y}$ of the expected response slightly decreases. Hence, there might be a negative trend; people born early in the year have a tendency of being drafted later.

1.3 To check whether the lottery is random, it is reasonable to use test statistics

$$T = \frac{\tilde{Y}(X_b) - \tilde{Y}(X_a)}{X_b - Xa},$$

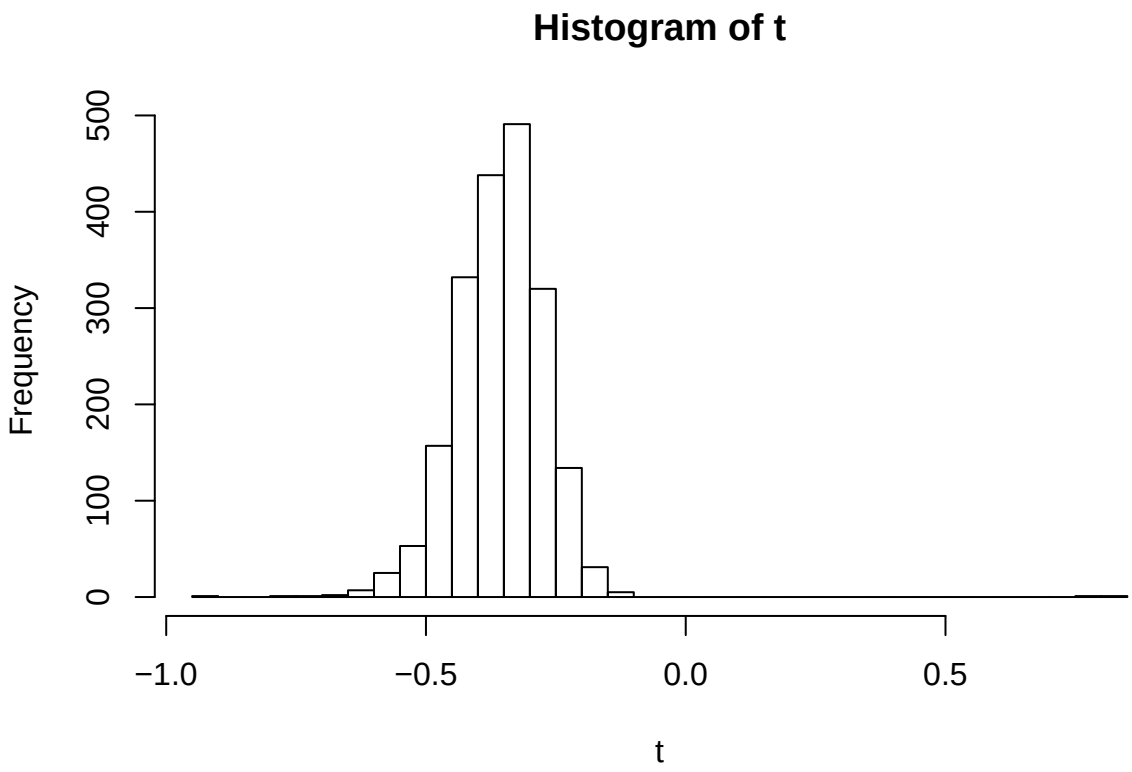
where $X_b = argmax_X \tilde{Y}, X_a = argmin_X \tilde{Y}$

If this value is significantly greater than zero, then there should be a trend in the data and the lottery is not random. Estimate the distribution of T by using a non-parametric bootstrap with $B = 2000$ and comment whether the lottery is random or not. What is the P-value of the test?

We have implemented the non-parametric bootstrap. First, we got the sample by sampling with replacement from the data, `lottery`. Then, we computed the expected response from the sample using `loess()` function and calculated the test statistics. The process is repeated 2000 times.

```
#X: Day_of_year
#Y: Draft_No
Tcal <- function(x, y){
    x.b <- x[which.max(y)]
    x.a <- x[which.min(y)]

    t <- (max(y) - min(y)) / (x.b - x.a)
```

```
    return(t)
}

#Non-parametric bootstrap
B <- 2000
t <- NULL
n <- nrow(lottery)
for(i in 1:B){
    data1 <- lottery[sample(n, n, replace=TRUE),]

    x <- data1$Day_of_year
    y <- loess(Draft_No~Day_of_year, data=data1)$fitted

    t[i] <- Tcal(x, y)
}
hist(t, 50)
```




```
#calculate p-value
ind <- which(t > 0)
pvalue <- 2 * ( length(ind)/B )
```

The figure above is the histogram of given test statistics value which is estimated by using the non-parametric bootstrap. We see that the lottery is not random and the p-value of the test is 0.002.

1.4 Implement a function depending on data and B and that tests the hypothesis H_0 : *Lottery is random* vs H_a : *Lottery is not random* by using a permutation test with statistics T and returns the p-value of this test. Test this function on our data with $B = 2000$

The permutation test is implemented in this step. We sample `x: Day_of_year` without replacement from the data and then use `loess()` function to get the estimated response. The test statistics as given above is computed and the whole process is repeated for 2000 times. The p-value of this test is returned from this function.

```
#Permutation test
myPermutation <- function(data, B){
    t <- numeric(B)
    n <- dim(data)[1]

    for(i in 1:B){
        x <- sample(data$Day_of_year, n, replace=FALSE)
        y <- loess(data$Draft_No ~ x)$fitted
        t[i] <- Tcal(x, y)
    }

    #calculate T from original data
    xt <- data$Day_of_year
    yt <- loess(Draft_No ~ Day_of_year, data=data)$fitted
    t.actual <- Tcal(xt, yt)

    #calculate p-value
    ind <- which(abs(t) > abs(t.actual))
    pvalue <- length(ind) / B
    return(pvalue)
}
per <- myPermutation(lottery, B=2000)
```

The p-value of this test is 0.1365. Assuming that the significance level for this test is chosen to be 0.05, we will see that the p-value is greater than α. We can not reject the null hypothesis and so we conclude that lottery is random.

1.5 Make a crude estimate of the power of the test constructed in step 4:

a. Generate (an obviously non-random) dataset with $n = 366$ observations by using same X as in the original data set and $Y(X) = min(0, max(\alpha x + \beta, 366))$ where $\alpha = 0.1$ and $\beta \sim N(183, sd = 10)$

b. Plug these data into the permutation test with $B = 200$ and note whether it was rejected

c. Repeat steps a)-b) for $\alpha = 0.2, 0.3, ..., 10$

What can you say about the quality of your test statistics considering the value of the power?

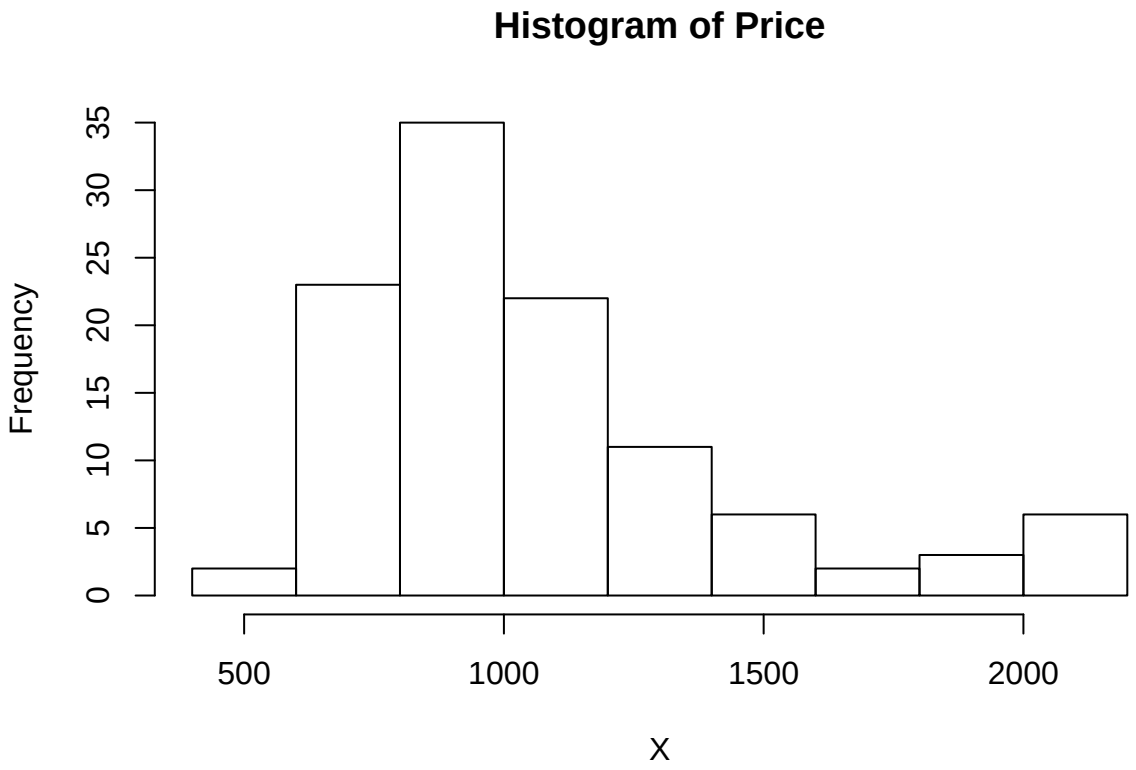
We assume that the significance level for this analysis here is equal to 0.05. A crude estimate of the power of this test is 1, which is computed from the ratio of the correct rejections. From this value, it can be said that the quality of test statistics is pretty good as p-values from every test we have got are all less than 0.05 and we assume that the data is non-random for all values of α.

We also tried to use some truly random data and found that the null hypotheses were not rejected. This also confirms the result we have got.

Assignment 2: Bootstrap, jackknife and confidence intervals

The data you are going to continue analyzing is the database of home prices in Albuquerque, 1993. The variables present are Price; SqFt – the area of a house; FEATS – number of features such as dishwasher, refrigerator and so on; Taxes – annual taxes paid for the house. Explore the file prices1.xls

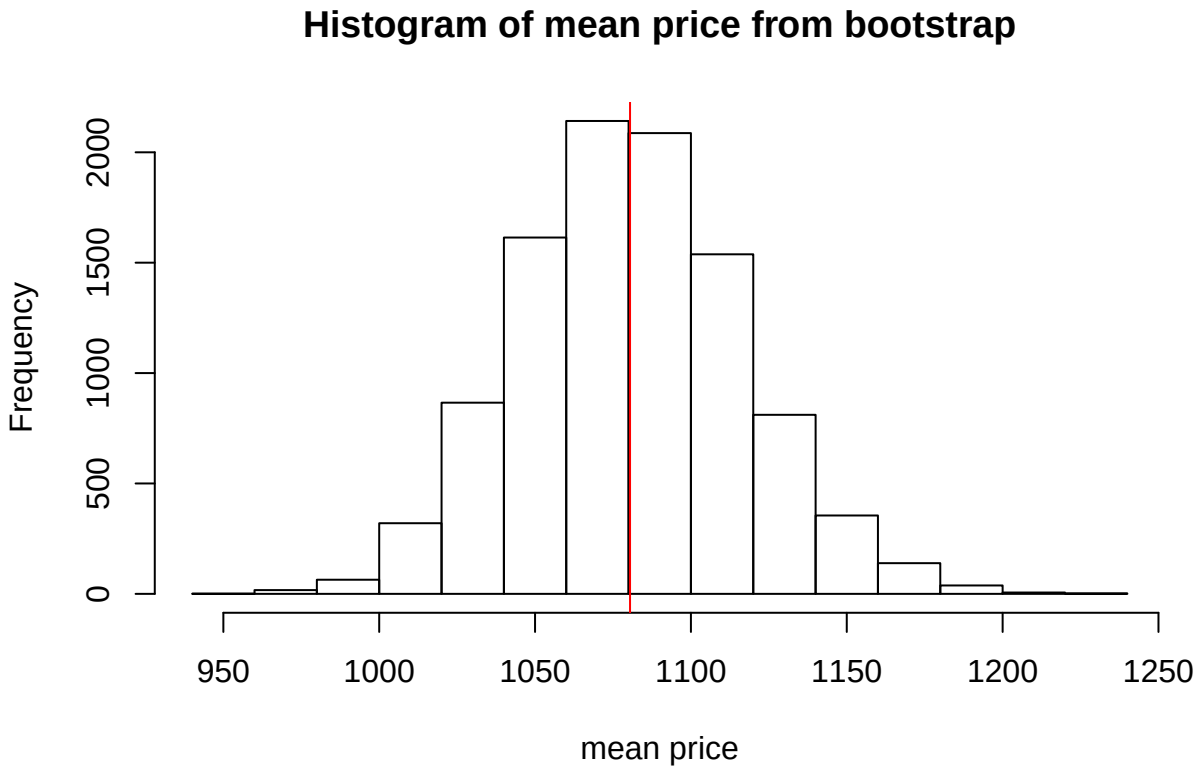
2.1 Plot the histogram of Price. Does it remind any conventional distribution? Compute the mean price.



The histogram shows that value of price is skewed to the right. The distribution of the sampled prices looks similar to some well-known ones, for example the χ^2-distribution, except for the heavy right tail.

The value of mean price is equal to 1080.4727273.

2.2 Estimate the distribution of the mean price of the house using bootstrap. Determine the bootstrap bias-correction and the variance of the mean price. Compute the 95% confidence interval for the mean price using bootstrap percentile, bootstrap BCa, and first-order normal approximation (Hint: use *boot(), boot.ci(), plot.boot(), print.bootci()*)




```
# mean price using bootstrap
print(mean(M.2.2$t))
```

```
## [1] 1080.695
```

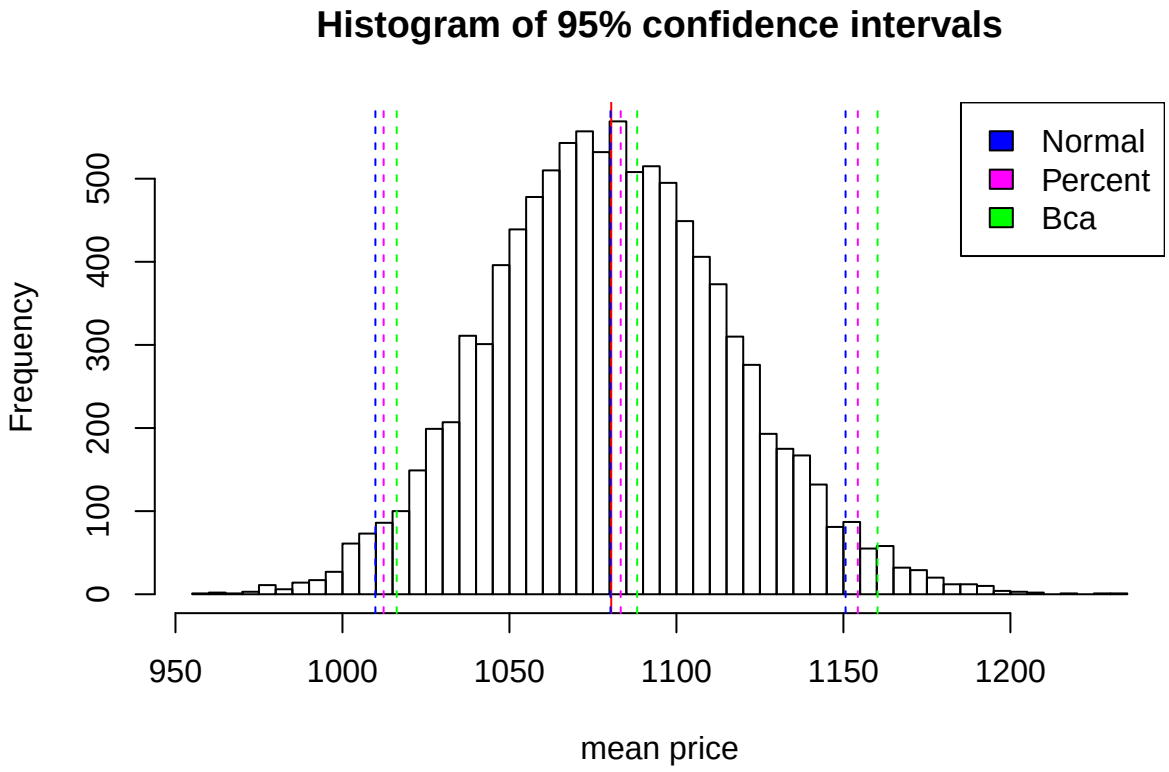
```
# bootstrap bias-correction
T1 <- 2 * w_hat - mean(M.2.2$t)
print(T1)
```

```
## [1] 1080.25
```

```
# bootstrap variance estimation
print(as.numeric(var(M.2.2$t)))
```

```
## [1] 1289.94
```

We have used non-parametric bootstrap to compute the mean price of the data. The mean of the original sample, $T(D)$, and the mean of the bootstrap samples, $\frac{1}{B}(T(D_1) + ... + T(D_B))$, are almost equal, so the bias correction T_1 is very similar, too.



No comment. See 2.4.

2.3 Estimate the variance of the mean price using the jackknife and compare it with the bootstrap estimate

```
# Jackknifed T mean
print(JT)
```

```
## [1] 1080.473
```

```
# Jackknifed T variance
print(VarT)
```

```
## [1] 1320.911
```

The Jackknifed T is the same as the sample mean and the variance of it, $V(\hat{T})_J$, is 1320.9110441. So, the bootstrap variance is a little smaller but it is also biased while the Jackknife distribution is centered around the mean value of the original sample because we only deleted 1 observation in each run.

2.4 Compare the confidence intervals obtained with respect to their length and the location of the estimated mean in these intervals.

We can see that the various conficende intervals are very similar to each other, but the normal seems to be the least biased and also has the smallest variance. (The sample mean of the bootstrap is very close to the first-order normal approximation and it can barely be seen.)

Appendix

Contribution

The code and most of the text for the first assignment are from Araya's report. For the second assignment, the code and most of the text are from Oscar.

R-code

```
lottery <- read.csv("/Users/lynn/Documents/LiU/732A38 Computational statistics/Lab5/lottery.csv")
#1
plot(lottery$Day_of_year, lottery$Draft_No, type="l", xlab="X: Day of year", ylab="Y: Draft No", col="da
fit <- loess(Draft_No~Day_of_year, data=lottery)
plot(lottery$Day_of_year, lottery$Draft_No, type="l", xlab="X: Day of year", ylab="Y: Draft No", col="da
lines(lottery$Day_of_year, fit$fitted, col="purple", lwd=2)
#X: Day_of_year
#Y: Draft_No
Tcal <- function(x, y){
    x.b <- x[which.max(y)]
    x.a <- x[which.min(y)]

    t <- (max(y) - min(y)) / (x.b - x.a)
    return(t)
}

#Non-parametric bootstrap
B <- 2000
t <- NULL
n <- nrow(lottery)
for(i in 1:B){
    data1 <- lottery[sample(n, n, replace=TRUE),]

    x <- data1$Day_of_year
    y <- loess(Draft_No~Day_of_year, data=data1)$fitted

    t[i] <- Tcal(x, y)
}
hist(t, 50)

#calculate p-value
ind <- which(t > 0)
pvalue <- 2 * ( length(ind)/B )
#Permutation test
myPermutation <- function(data, B){
    t <- numeric(B)
    n <- dim(data)[1]

    for(i in 1:B){
        x <- sample(data$Day_of_year, n, replace=FALSE)
        y <- loess(data$Draft_No ~ x)$fitted
        t[i] <- Tcal(x, y)
    }
```

```
    #calculate T from original data
    xt <- data$Day_of_year
    yt <- loess(Draft_No ~ Day_of_year, data=data)$fitted
    t.actual <- Tcal(xt, yt)

    #calculate p-value
    ind <- which(abs(t) > abs(t.actual))
    pvalue <- length(ind) / B
    return(pvalue)
}
per <- myPermutation(lottery, B=2000)
#5
#crude estimate of the power
alpha <- seq(0.1,10,0.1)
beta <- rnorm(366, mean=183, sd=10)
result <- NULL
y <- NULL
data2 <- lottery
for(j in 1:length(alpha)){
    x <- data2$Day_of_year
    for(i in 1:length(x)){
        y[i] <- max(0, min(alpha[j]*x[i] + beta[j], 366))
    }
    data2$Draft_No = y
    result[j] <- myPermutation(data2, B=200)
}
reject.index <- which(result < 0.05) #correct rejection
power <- length(reject.index)/100
price1 <- read.csv("/Users/lynn/Documents/LiU/732A38 Computational statistics/Lab5/prices1.csv")
X <- price1$Price
hist(X, main="Histogram of Price")
w_hat <- mean(X)
library(boot)
stat <- function(data, indices) mean(data[indices], 0, FALSE)
set.seed(12345)
M.2.2 <- boot(data = X, statistic = stat, R = 10E3)

hist(M.2.2$t, main="Histogram of mean price from bootstrap", xlab="mean price")
abline(v = w_hat, col = "red")
# mean price using bootstrap
print(mean(M.2.2$t))

# bootstrap bias-correction
T1 <- 2 * w_hat - mean(M.2.2$t)
print(T1)

# bootstrap variance estimation
print(as.numeric(var(M.2.2$t)))
hist(M.2.2$t, main="Histogram of 95% confidence intervals", xlab="mean price", 50)
CIs <- boot.ci(boot.out = M.2.2)
# boot:::plot.boot(x = M.2.2)
# boot:::print.boot(x = M.2.2)
abline(v = w_hat, col = "red")
```

```r
abline(v = CIs$normal[2:3], col = "blue", lty = "dashed")
abline(v = mean(CIs$normal[2:3]), col = "blue", lty = "dashed")
abline(v = CIs$percent[4:5], col = "magenta", lty = "dashed")
abline(v = mean(CIs$percent[4:5]), col = "magenta", lty = "dashed")
abline(v = CIs$bca[4:5], col = "green", lty = "dashed")
abline(v = mean(CIs$bca[4:5]), col = "green", lty = "dashed")
legend(x = "topright", legend = c("Normal", "Percent", "Bca"), fill = c("blue", "magenta", "green"))
T_j <- rep(NA_real_, length(X))
r <- length(X)
for(i in seq_along(X)) {
  Y_j <- X[-i]
  T_j[i] <- mean(Y_j)
}

T_bar <- sum(T_j) / r
T_star_j <- r*mean(X) - (r - 1) * T_j
JT <- sum(T_star_j) / r
VarT <- sum((T_star_j - JT)^2) / (r * (r - 1))
# Jackknifed T mean
print(JT)

# Jackknifed T variance
print(VarT)
## NA
```